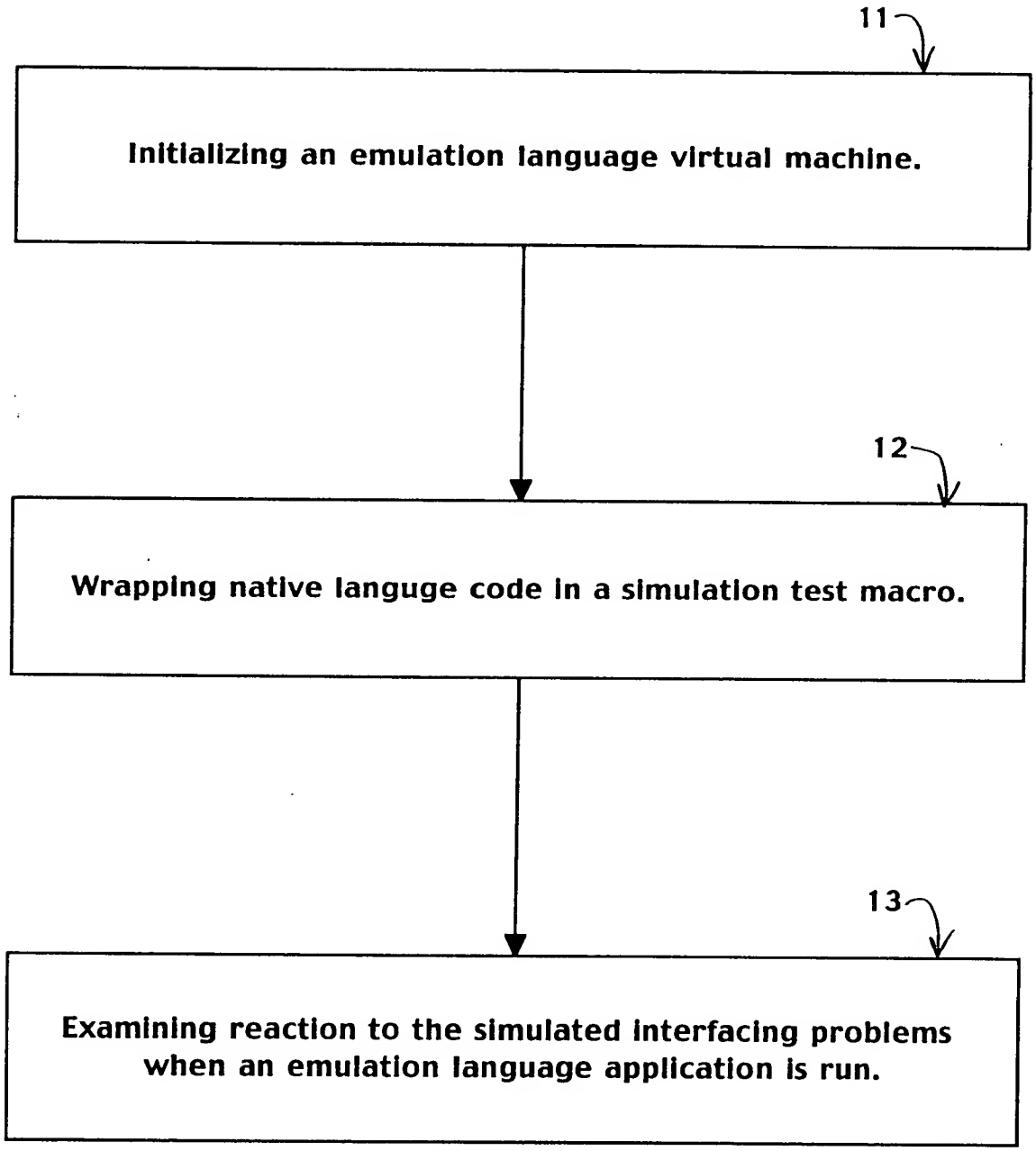
10
11
Initializing an emulation language virtual machine.
12
Wrapping native languge code in a simulation test macro.
13
Examining reaction to the simulated interfacing problems when an emulation language application is run.

FIG 1A

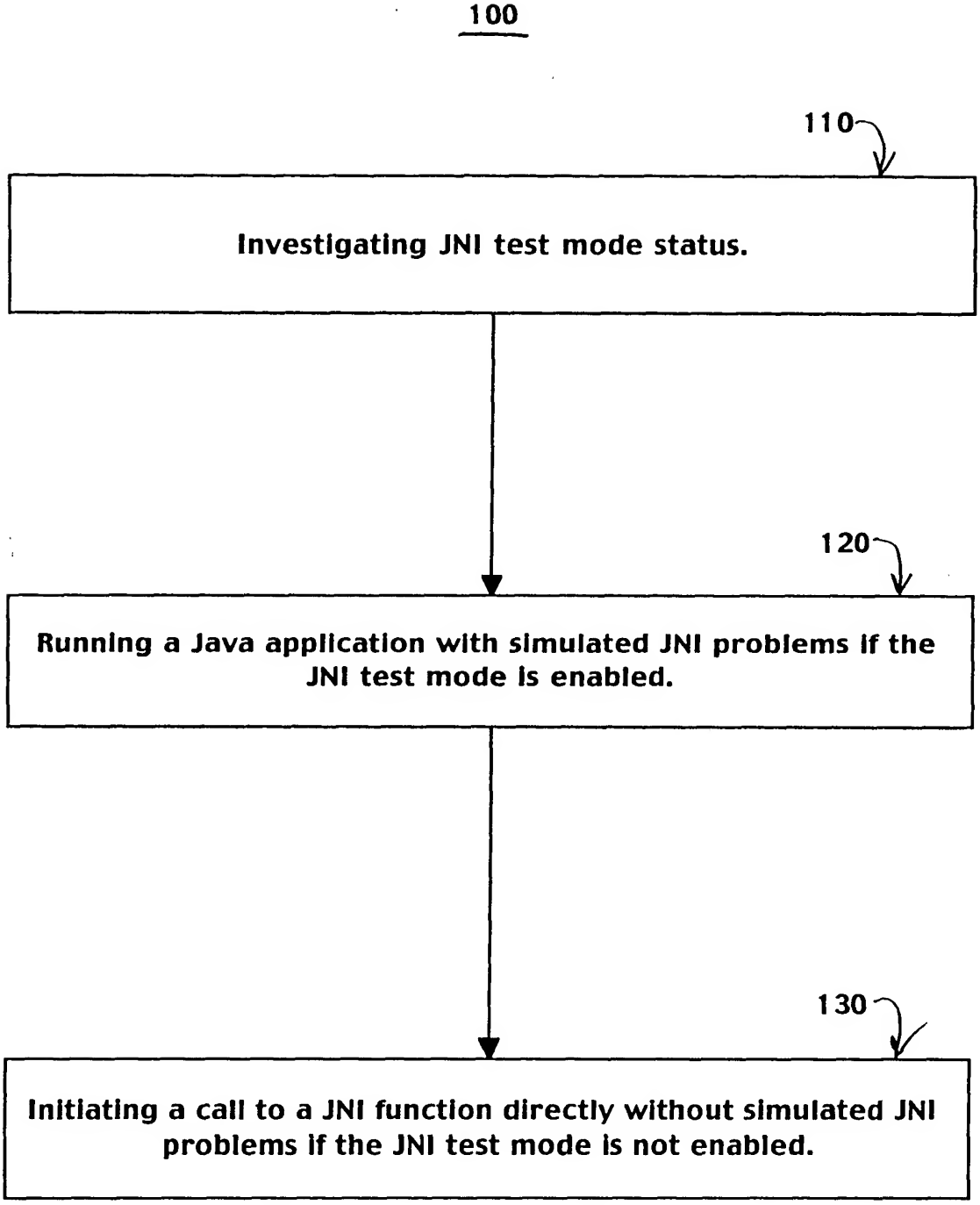
100
110
Investigating JNI test mode status.
120
Running a Java application with simulated JNI problems if the JNI test mode is enabled.
130
Initiating a call to a JNI function directly without simulated JNI problems if the JNI test mode is not enabled.

FIG 1B

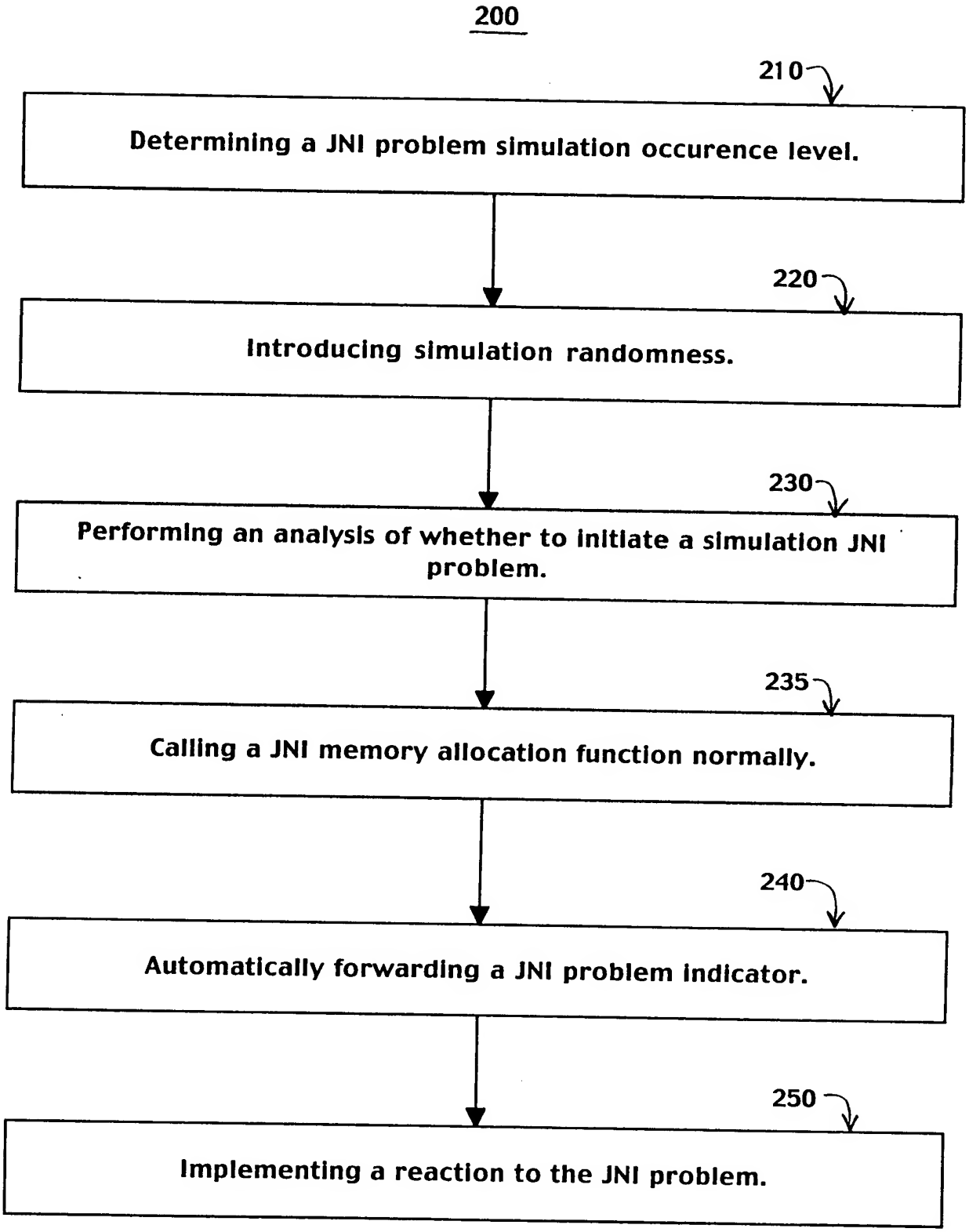
200
210
Determining a JNI problem simulation occurence level.
220
Introducing simulation randomness.
230
Performing an analysis of whether to initiate a simulation JNI problem.
235
Calling a JNI memory allocation function normally.
240
Automatically forwarding a JNI problem indicator.
250
Implementing a reaction to the JNI problem.

FIG 2

300

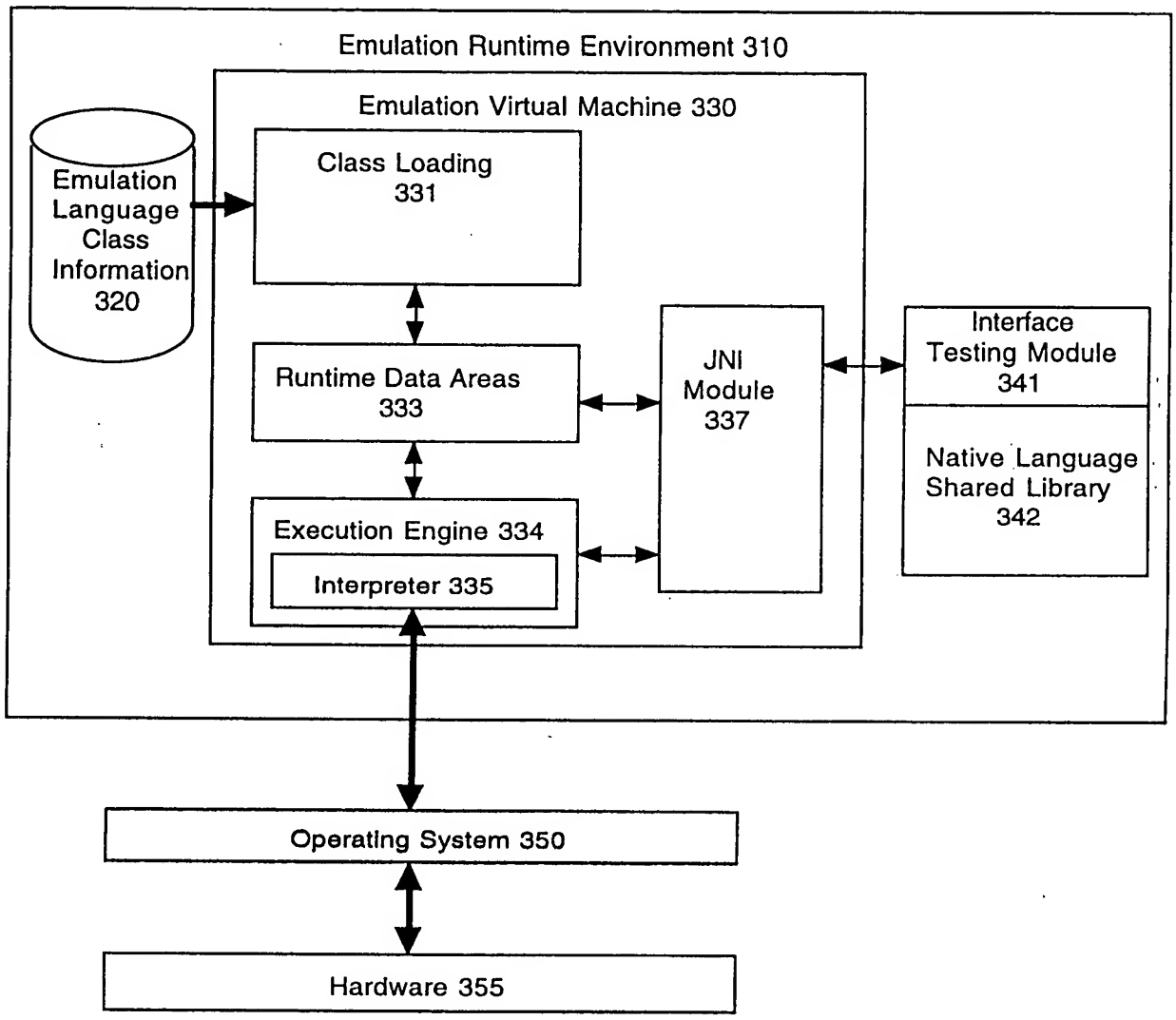
Emulation Runtime Environment 310
Emulation Virtual Machine 330
Emulation Language Class Information 320
Class Loading 331
Runtime Data Areas 333
Execution Engine 334
Interpreter 335
JNI Module 337
Interface Testing Module 341
Native Language Shared Library 342
Operating System 350
Hardware 355

FIG 3

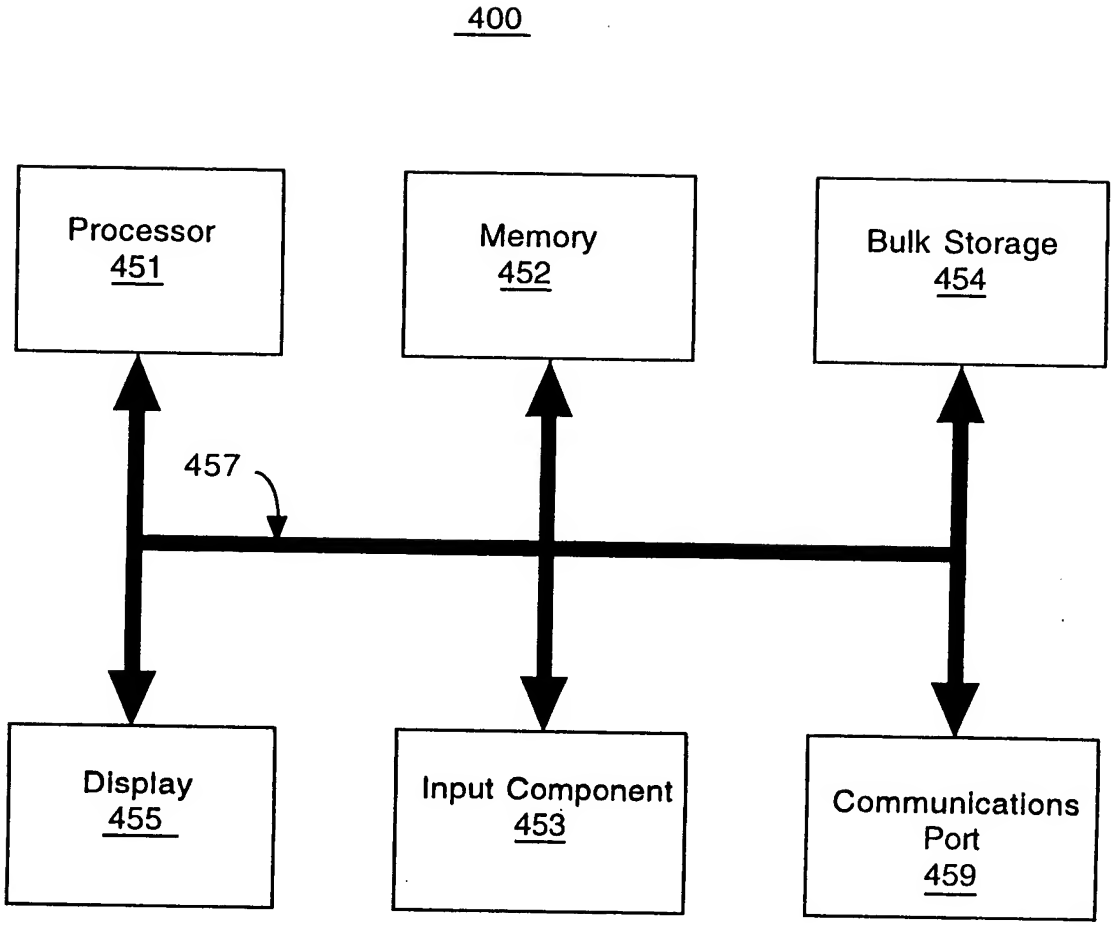
400
Processor
451
Memory
452
Bulk Storage
454
457
Display
455
Input Component
453
Communications
Port
459

FIG 4